

## BUNDESREPUBLIK DEUTSCHLAND

27.04.2004

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)



REC'D 14 MAY 2004

WIPO

PCT

**Prioritätsbescheinigung über die Einreichung  
einer Patentanmeldung**

**Aktenzeichen:** 103 01 310.5

**Anmeldetag:** 15. Januar 2003

**Anmelder/Inhaber:** Continental Teves AG & Co oHG,  
Frankfurt am Main/DE

**Bezeichnung:** Speicherarchitektur für Kraftfahrzeugrechner

**IPC:** G 06 F 12/16

**Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.**

München, den 09. Januar 2004  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

Hoß

Continental Teves AG & Co. oHG

15.01.2003

P 10609

GP/BR/ad

Dr. Voß

Dr. Traskov

Dr. Kirschbaum

### **Speicherarchitektur für Kraftfahrzeugrechner**

Die Erfindung betrifft ein Verfahren zur Überwachung und zur Fehlerkorrektur von Speicher in sicherheitskritischen Applikationen, welches insbesondere in einem elektronischen Kraftfahrzeugsteuergerät durchgeführt wird.

Elektronische Steuergeräte für Kraftfahrzeugbremsen übernehmen im Zuge der technischen Entwicklung immer mehr Funktionen des Bremssystems. Wurde früher lediglich die ABS-Funktion elektronisch gesteuert und geregelt, wird in heutigen "by-wire" Bremsanlagen die vollständige Bremsfunktion vom elektronischen Steuergerät kontrolliert. Daher besteht ein zunehmender Bedarf an elektronischen Kraftfahrzeugsteuergeräten mit hoher Zuverlässigkeit.

Üblicherweise umfassen bekannte elektronische Kraftfahrzeugsteuergeräte zur Bewältigung der komplexen Funktionen programmgesteuerte Mikroprozessorsysteme. Es ist weiterhin bekannt, dass die Zuverlässigkeit der gattungsgemäßen Mikroprozessorsysteme verbessert wird, wenn bei der Speicherung von Daten in einem RAM-Speicher Prüfdaten erzeugt werden.

Die Zuverlässigkeit eines Kraftfahrzeugrechnersystems lässt sich gemäß der DE 101 09 449 auch dadurch verbessern, dass beim Lesen von Flash-Speichern durch den Mikroprozessor für jede Datenzeile Paritätsbits im gleichen Speicherbaustein

- 2 -

oder in einem separaten Speicherbaustein abgelegt werden. Während des Speicherzugriffs werden ebenfalls Paritätsbits erzeugt und zum Zwecke der Fehlerüberprüfung mit den gespeicherten Prüfdaten verglichen.

Eine zeilenweise Absicherung von Daten in einem Datenspeicher durch Speicherung von Paritätsbits erlaubt zwar das Erkennen von Einzelfehlern für kleine Blockgrößen (Halbworte/ Worte), ist jedoch vergleichsweise speicherplatzaufwendig. Daher wird üblicherweise in abgesicherten mehrkernigen Mikroprozessorssystemen ein zusätzlicher Paritätsspeicher geschaffen, in dem lediglich die Prüfdaten gespeichert werden. Dies bietet gegenüber einem vollredundanten Speicher den Vorteil, dass Chipfläche eingespart werden kann.

Es ist bekannt, Prüfdaten mittels Fehlerkorrekturcodes wie Hamming Code oder Berger Code auf dem Speicherchip / Speichercore vorzusehen, um

- transiente Fehler zu korrigieren.
- die garantierte Lebensdauer eines Produktes zu erhöhen.
- durch das Maskieren von Fertigungsfehlern die Ausbeute zu erhöhen.

Vorzugsweise werden die Prüfdaten nach einem CRC- oder ECC-Verfahren erzeugt.

Bei diesen bekannten Verfahren und Bauelementen ist die Fehlerkorrektur im Speichercore integriert, so dass Fehler im Speicherwrapper und auf den Adress- und Datenleitungen zum Core hin nicht erkannt bzw. korrigiert werden können.

- 3 -

Außerdem wird die Adresse an sich nicht mit in die Fehlererkennung einbezogen.

Unter dem Begriff Rechnersystem werden ganz allgemein einzelne oder auch vernetzte Computersysteme wie z.B. Mikrocontroller verstanden, welche neben einer Zentralrecheneinheit (CPU) zusätzlich Speicher und Ein-/Ausgabefunktionen umfassen. Diese Systeme können "einkernig" oder insbesondere "mehrkernig" ausgeführt sein, wobei bei einem mehrkernigen System die Rechnersysteme zwei oder mehrere Zentralrechenheiten umfassen.

Um auch diese Fehler sicher erkennen zu können, wird insbesondere in mehrkernigen Mikroprozessoren eine Architektur benutzt, die einen vom Programmspeicher vollkommen getrennten Paritätsspeicher verwendet. Das bedeutet nun aber einen erheblichen Mehraufwand bezüglich Fläche und damit Kosten für einen zweiten Wrapper mit Ladungspumpe.

Beim insbesondere als Flash-Memory ausgeführten Applikationsspeicher können nur ganze Segmente gelöscht und programmiert werden. Ein klein segmentierter Flashspeicher benötigt eine erheblich größere Fläche verglichen mit einem Flashspeicher mit größeren Segmenten. Deshalb ist die Größe der Segmente nach unten hin von den Herstellern begrenzt. Da einzelne Programm- und Datenteile unabhängig voneinander programmiert und gelöscht werden müssen (sowohl die Daten als auch das zugehörige Parity) und die kleinste Segmentgröße des Parity beschränkt ist, ergeben sich sehr große Bereiche ungenutzten Speichers.

- 4 -

Es wird ein Verfahren vorgeschlagen, das die Fehlerkorrektur zum Empfänger der Daten hin verschiebt und somit eine Überwachung und Fehlerkorrektur sowohl des Datenfeldes als auch der Speicherwrapper mit Adressdekodierung und der Daten-/Adressleitungen zuläßt. Weiterhin werden die Adressdaten mit zur Berechnung der Fehlererkennungskodes herangezogen.

**Beschreibung der Erfindung:**

**Überprüfung der Adressen:**

Adressdaten werden mit in die Berechnung der Checkbits einbezogen, d.h. sowohl die Dekodierung / Fehlerkorrektur als auch die Codegenerierung basiert auf Daten **und** Adressen (siehe Abb. 1). Damit können eventuelle Fehler in der Adressdekodierlogik erkannt werden (Fehler 4 in Abb. 1). Durch die Berücksichtigung der Adressen entsteht kein zusätzlicher Speicheraufwand. Nach einer bevorzugten Ausführungsform der Erfindung werden die Prüfdaten nach der Methode der Hammingkode-Generierung mit einer Hammingdistanz von insbesondere etwa 4 erzeugt. Hierdurch ist es z. B. möglich mit 8 Bit Prüfdaten 120 Datenbits zu kodieren. Da üblicherweise Datenbreiten in Zweierpotenzen benutzt werden, stehen genügend Bits für die Einbeziehung der Adressen zur Verfügung. Es ist nicht immer sinnvoll, die Adressen zu korrigieren, da die Daten, die zu der fehlerhaften Adresse gehören und nicht angefordert wurden, ausgelesen wurden.

**Überprüfung / Korrektur der Daten- und Adressleitungen:**

Durch die Verlagerung der Überprüfungs- und Korrekturlogik hin zum Datenempfänger werden die Pfade zwischen Speicher und Datenempfänger mit überprüft und gegebenenfalls korrigiert (Fehler 1-3 in Abb. 1).

Überprüfung der Korrekturlogik durch eine zweite Überprüfungseinheit:

Die Korrektur eventueller Fehler wird durch eine zweite Überprüfungseinheit überwacht. Dabei werden aus den korrigierten Daten und den Adressen der zweiten CPU Checkbits berechnet und mit den (ev. korrigierten) Checkbits vom Speicher verglichen (siehe Abb. 2). Damit kann das Verfahren für die Teves Dual Core Architektur verwendet werden.

Das hier vorgeschlagenen Verfahren ist besonders effektiv für externen Speicher und für separaten Speicher innerhalb eines Multi-Chip-Moduls (MCM), kann aber auch für On-Chip-Speicher erhebliche Vorteile bringen.

Das erfindungsgemäße Rechnersystem ist vorzugsweise Bestandteil eines elektronischen Kraftfahrzeugsteuergerätes, insbesondere eines elektronischen Reglers (ECU), welcher mit einem hydraulischen Bremsensteuergerät (HCU) steckbar zu einem blockförmigen Verbund zusammengefügt ist. Die Erfindung betrifft daher auch die Verwendung des weiter oben beschriebenen Rechnersystems in elektronischen Steuergeräten für Kraftfahrzeuge, insbesondere in elektronischen Kraftfahrzeugbremssystemen.

Vorteile:

- o Bestehende Lösungen überprüfen und korrigieren nur Fehler im Datenarray (Fehler 5 und 6 in Abb.1).
- o Das vorgeschlagene Verfahren erlaubt die Überprüfung der Adressdekodierlogik (Fehler 4 in Abb. 1)

- 6 -

- o Das vorgeschlagene Verfahren erlaubt die Überprüfung und mögliche Fehlerkorrektur der Daten- und Adressleitungen (Fehler 1-3 in Abb. 1)
- o Mit einer zweiten Fehlerdetektionslogik kann die eigentliche Fehlerkorrektur und -erkennung überprüft werden.
- o Verglichen mit der zur Zeit in Kraftfahrzeugbremssystemen verwendeten Architektur kann der separate Paritätspeicher inklusive Wrapper und Ladungspumpe eingespart werden. Das Problem der kleinen Segmentierung mit großen Bereichen ungenutzten Speichers wird damit vermieden.

#### Zusammenfassung

- Erweiterung der Fehlererkennung auf Übertragungswege (besonders wichtig bei externen Komponenten mit störfälligen langen Verbindungen zum Rechenkern).
- Erweiterung der Fehlererkennung auf den Adressdecoder. Das ist besonders dann vorteilhaft, wenn nur ein einziger Adressdecoder verwendet werden soll.
- Erweiterung der Fehlererkennung auf die Fehlerkorrektur und -erkennung.
- Kein Performanceverlust gegenüber bestehenden Lösungen

**Patentansprüche**

1. Verfahren zur Erkennung und/oder Vermeidung von Speicherzugriffsfehlern in einem Rechnersystem mittels einer ersten Fehlererkennungseinrichtung (2), insbesondere umfassend mindestens zwei Rechnerkerne, bei dem innerhalb des Speichers (3) neben den abzusichernden Daten (6) zusätzlich Prüfdaten (7) abgelegt werden, welche unter Verwendung der abzusichernden Daten erzeugt wurden, dadurch **gekennzeichnet**, dass bei der Prüfdatengenerierung Daten mit den jeweils zugehörigen Adressen dieser Daten verknüpft und diese im gleichen Speicher, insbesondere im gleichen Datensegment, aus dem die zu überprüfenden Daten stammen, gespeichert werden.
2. Verfahren nach Anspruch 1, dadurch **gekennzeichnet**, dass bei der Generierung der Prüfdaten mehrere Zeilen des Speichers verknüpft werden.
3. Verfahren nach Anspruch 1 oder 2, dadurch **gekennzeichnet**, dass sowohl bei der Speicher-Dekodierung, der Fehlerkorrektur als auch bei der Prüfdatengenerierung Daten gemeinsam mit deren zugehörigen Adressen verarbeitet werden.
4. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch **gekennzeichnet**, dass die erste Fehlererkennungseinrichtung durch eine weitere Überprüfungseinheit (8) überprüft wird.

5. Verfahren nach Anspruch 4, dadurch **gekennzeichnet**, dass Adressdaten, welche an einem Adressbus zwischen dem weiteren Rechnerkern (9) und der weiteren Überprüfungseinheit (8) anliegen, von Überprüfungseinheit (8) zur Erzeugung von Vergleichsprüfdaten (16) verarbeitet werden.
6. Verfahren nach Anspruch 4 oder 5, dadurch **gekennzeichnet**, dass die weitere Überprüfungseinheit Vergleichsprüfdaten aus Daten und Adressen erzeugt, welche mit Prüfdaten der ersten Fehlererkennungseinrichtung oder mit Prüfdaten des Speichers, welcher mit der ersten Fehlererkennungseinrichtung verbunden ist, verglichen werden, wobei die verwendeten Daten vom Datenbereich der ersten Rechnereinheit stammen.
7. Verfahren nach mindestens einem der Ansprüche 1 bis 6, dadurch **gekennzeichnet**, dass die Daten, Prüfdaten und Adressen, welche zwischen der ersten Fehlererkennungseinrichtung und dem Applikationsspeicher übertragen werden, mittels getrennter Busleitungen (11, 12, 13) übertragen werden.
8. Verfahren nach mindestens einem der Ansprüche 1 bis 7, dadurch **gekennzeichnet**, dass zumindest die Daten und die Prüfdaten, welche zwischen der weiteren Überprüfungseinheit (8) und der erste Fehlererkennungseinrichtung (2) und/oder dem Applikationsspeicher übertragen werden, mittels getrennter Busleitungen (14, 15) übertragen werden.

9. Elektronische Schaltungsanordnung, insbesondere zur Durchführung des Verfahrens nach mindestens einem der Ansprüche 1 bis 8, dadurch **gekennzeichnet**, dass ein erster Rechnerkern (1) mit einer ersten Fehlererkennungseinrichtung (2) verbunden ist, welche mit einem Datenspeicher (3) verbunden ist, wobei die Fehlererkennungseinrichtung einen Prüfdatengenerator (4), eine Buslogikeinrichtung (5) sowie einen Korrekturblock (6) umfasst.
10. Schaltungsanordnung nach Anspruch 9, dadurch **gekennzeichnet**, dass die Fehlererkennungseinrichtungen nach der Erfindung nicht in der Speicherlogik (10) implementiert sind.
11. Schaltungsanordnung nach Anspruch 9 oder 10, dadurch **gekennzeichnet**, dass der Speicher (3) nur segmentweise lösbar ist und insbesondere zur Speicherung von Applikationssoftware dient.
12. Schaltungsanordnung nach mindestens einem der Ansprüche 9 bis 11, dadurch **gekennzeichnet**, dass der Speicher und der Rechnerkern nicht auf dem gleichen Chip angeordnet sind.
13. Schaltungsanordnung nach mindestens einem der Ansprüche 9 bis 11, dadurch **gekennzeichnet**, dass der Speicher und der Rechnerkern auf dem gleichen Chip angeordnet sind.

- 10 -

14. Schaltungsanordnung nach mindestens einem der Ansprüche 9 bis 13, dadurch **gekennzeichnet**, dass ein oder mehrere weitere Rechnerkerne (9) mit dem ersten Rechnerkern in einem Bauelement zusammengefasst sind, wobei der/die weiteren Rechnerkerne nicht über deren eigenen Bus mit einem dem/den weiteren Rechnerkernen zugeordneten Applikationsspeicher verbunden sind, sondern für deren Verarbeitung den mit dem ersten Rechnerkern verbundenen Applikationsspeicher nutzen.
15. Schaltungsanordnung nach mindestens einem der Ansprüche 9 bis 14, dadurch **gekennzeichnet**, dass der mindestens eine weitere Rechnerkern Speicherzugriffe über die erste Fehlererkennungseinrichtung durchführt.
16. Schaltungsanordnung nach Anspruch 15, durch **gekennzeichnet**, dass die durch die erste Fehlererkennungseinrichtung an den/die weiteren Rechnerkerne übermittelten Daten überprüft und im Fehlerfall, falls eine Fehlerkorrektur möglich ist, korrigiert sind.

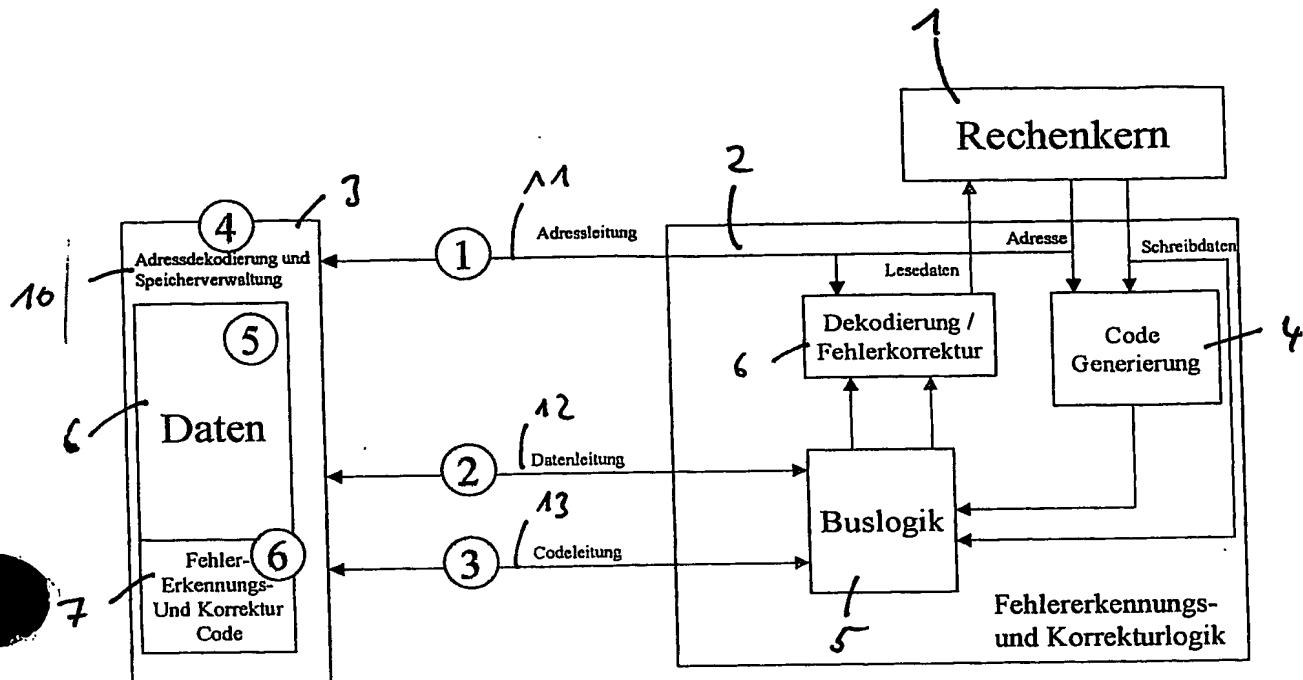


Fig. 1

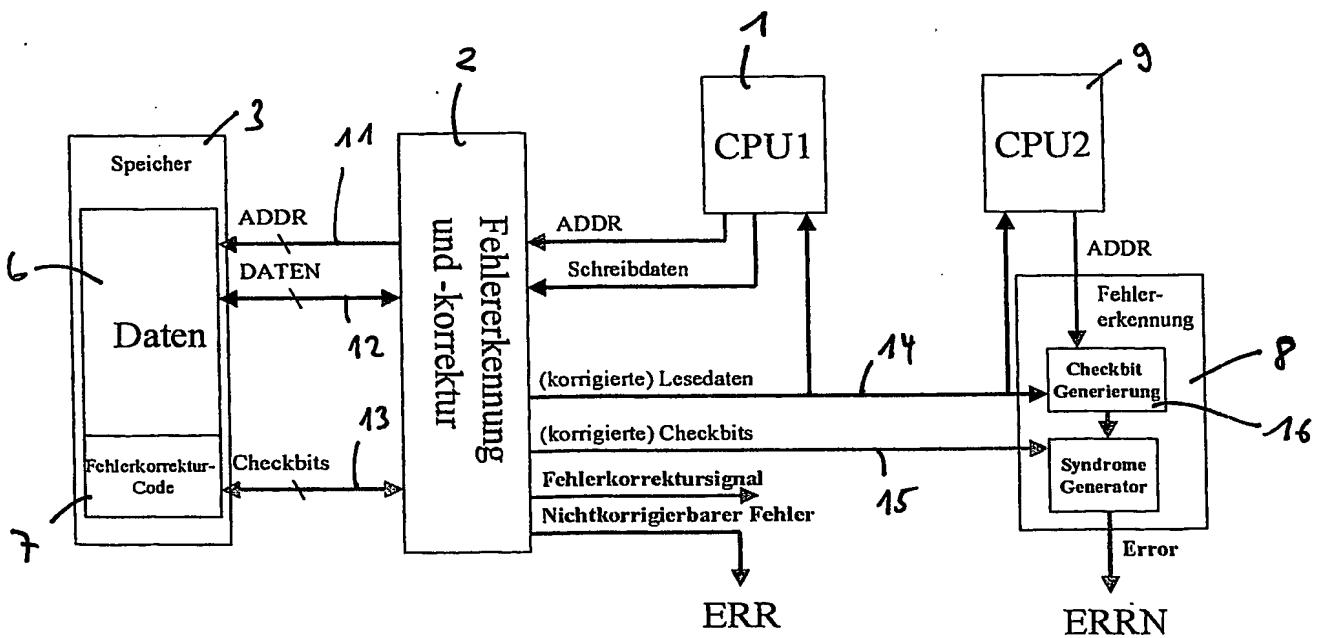


Fig. 2

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

**BLACK BORDERS**

**IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

**FADED TEXT OR DRAWING**

**BLURRED OR ILLEGIBLE TEXT OR DRAWING**

**SKEWED/SLANTED IMAGES**

**COLOR OR BLACK AND WHITE PHOTOGRAPHS**

**GRAY SCALE DOCUMENTS**

**LINES OR MARKS ON ORIGINAL DOCUMENT**

**REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

**OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**